

# From Point to Set: Extend the Learning of Distance Metrics

Pengfei Zhu, Lei Zhang\*, Wangmeng Zuo, David Zhang  
The Hong Kong Polytechnic University  
Hong Kong China

cspzhu, cslzhang@comp.polyu.edu.hk

## Abstract

*Most of the current metric learning methods are proposed for point-to-point distance (PPD) based classification. In many computer vision tasks, however, we need to measure the point-to-set distance (PSD) and even set-to-set distance (SSD) for classification. In this paper, we extend the PPD based Mahalanobis distance metric learning to PSD and SSD based ones, namely point-to-set distance metric learning (PSDML) and set-to-set distance metric learning (SSDML), and solve them under a unified optimization framework. First, we generate positive and negative sample pairs by computing the PSD and SSD between training samples. Then, we characterize each sample pair by its covariance matrix, and propose a covariance kernel based discriminative function. Finally, we tackle the PSDML and SSDML problems by using standard support vector machine solvers, making the metric learning very efficient for multi-class visual classification tasks. Experiments on gender classification, digit recognition, object categorization and face recognition show that the proposed metric learning methods can effectively enhance the performance of PSD and SSD based classification.*

## 1. Introduction

How to select a proper distance metric is a key problem in pattern classification, while the optimal distance metric for a specific pattern classification task depends on the underlying data structure and distributions. In recent years, it has been increasingly popular to learn a desired distance metric from the given training samples in many visual classification tasks, such as face/action/kinship verification [14], visual tracking [18], and image retrieval [1]. Metric learning methods can be categorized into unsupervised [9], semi-supervised [3] and supervised ones [14, 18, 1], according to the availability of the class labels of training samples.

In general, metric learning aims to learn a valid distance

metric, measured by which the samples from the positive sample pair (i.e., samples with the same class label or similar samples) could be as close as possible, while the samples from the negative sample pair (i.e., samples with the different class labels or dissimilar samples) could be as far as possible. Positive/negative sample pairs can be generated from the  $K$  nearest neighbors as in Large Margin Nearest Neighbor (LMNN) [31], Neighborhood Components Analysis (NCA) [13], or from the given sample pairs in verification as in Logistic Discriminative Metric Learning (LDML) [14], or from side information with some prior knowledge as in Information Theoretic Metric Learning (ITML) [10]. In some cases, only positive pairs are used in metric learning [14]. In [27], metric learning is formulated as a kernel classification model and the relations with LMNN and ITML are discussed. Metric learning algorithms have also been developed for multi-task learning [24], multiple instance learning [15] and nonlinear metrics [19].

Currently, almost all the metric learning methods focus on the learning of a point-to-point distance (PPD) metric in couple with the nearest neighbor classifier (NNC). In many computer vision tasks (e.g., face recognition), however, we need to measure the distance between an image (i.e., a point) and an image set (i.e., a point set). In video based recognition tasks [29] or multi-view object recognition [20], we even need to measure the distance between two image sets. Therefore, it is highly desired to design effective point-to-set distance (PSD) and set-to-set distance (SSD) metric learning methods. Unfortunately, many PPD metric learning methods cannot be readily applied to PSD and SSD based classification.

A set is often modeled as a hull, a convex hull (CH), or an affine hull (AH), and PSD can then be defined as the distance from a point to this hull. Correspondingly, the nearest subspace classifier (NSC), nearest convex hull classifier (NCH) [26], and nearest convex affine classifier (NAH) [26] are proposed for PSD based classification. In [6], a set is modeled as a bounding hyperdisk (the set formed by intersecting their affine hull and their smallest bounding hypersphere), and a nearest hyperdisk classifier (NHD) is pro-

\*Corresponding author

posed for classification [6]. Given a query sample, those PSD based classifiers (NSC, NCH, NAH and NHD) compute its distance to each class, i.e., the PSD between the query samples and the set of templates of this class, and classify it to the class with the minimal point-to-set distance. In [30], an image to class distance is learned in a multi-task way by considering each class as one task. In [36], an image to class distance is defined by minimizing the distance over all possible object configurations and all possible object matchings, and then the distance function parameters are learned. The work in [30] and [36] both focus on a special image to class distance rather than a general point to set distance.

The calculation of SSD also depends on the means to model a set. In [5], by modeling each set as a CH/AH, the CH/AH based image set distance (CHISD/AHISD) is defined. In [16], sparsity is imposed on the AH model and a sparse approximation nearest points (SANP) method is proposed for image set classification. In [35], a regularized affine hull (RAH) is proposed to model a set, and the SSD is defined between two RAHs. In [34], each set is represented by a linear subspace and the angles between two subspaces are utilized to measure the similarity of two sets. The method in [20] employs canonical correlation to measure the similarity between two sets. In [29], an image set is modeled as a manifold and a manifold-to-manifold distance (MMD) is proposed. After calculating the distance from the query set to each template set, those SSD based classifiers classify the query set to the class with the minimal set-to-set distance. To introduce discriminative information to SSD, projection matrix is learned in a large margin manner, e.g., discriminative canonical correlation (DCC) [20] and manifold discriminant analysis (MDA) [28]. In [32], a set based discriminative ranking model is proposed by iterating between SSD finding and discriminative feature space projection.

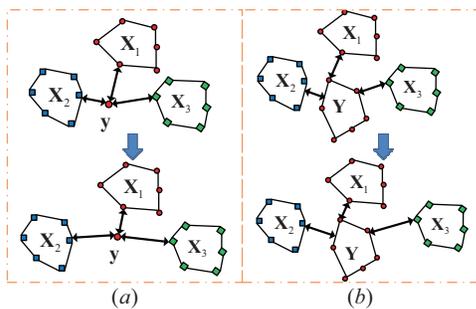


Figure 1. PSD (left) and SSD (right) Metric learning

Inspired by the success of metric learning in PPD based classification, the performance of PSD and SSD based classification can also be boosted by metric learning. As shown in the upper part of Fig. 1(a), the query image  $y$  (represented as a red dot) has the same class label as template set

$X_1$  (represented as a red hull) but it will be misclassified since it has a closer PSD to set  $X_2$ . If a proper metric learning method can be developed, it is possible that with the new distance metric, the PSD between  $y$  and  $X_1$  is smaller than that between  $y$  and  $X_2$ , and consequently  $y$  can be correctly classified, as shown in the bottom part of Fig. 1(a). Similar anticipation goes to the metric learning of SSD based classification, as illustrated in Fig. 1(b), where the query set  $Y$  can be correctly classified with some proper SSD based distance metric.

With the above considerations, in this paper we propose two novel metric learning models, PSD metric learning (PSDML) and SSD metric learning (SSDML), to enhance the performance of PSD and SSD based classification. One image (or image set) and one similarly labeled image set construct a positive pair, while one image (or image set) and one differently labeled set construct a negative pair. Then the PSDML and SSDML problems are formulated as a sample pair classification problem. Each sample pair is characterized by the covariance matrix of its two samples, and a covariance kernel is introduced. A discriminative function is then proposed for sample pair classification, and finally the PSDML and SSDML can be solved by using an SVM model. The proposed PSDML and SSDML methods can effectively improve the performance of PSD and SSD based classification, and are much more efficient than state-of-the-art metric learning methods.

The main abbreviations used in this paper are summarized in the following Table 1.

PPD	point to point distance
PSD	point to set distance
SSD	set to set distance
PSDML	point to set distance metric learning
SSDML	set to set distance metric learning

## 2. Set based distances

Before distance metric learning, we need to first define how the distance is measured. In this section, we describe how an image set is modeled, and how the corresponding point-to-set and set-to-set distances are defined.

### 2.1. Image set model

An image set is usually represented by a hull, i.e., a subspace spanned by all the available samples in the set. The hull of a set of samples  $D = [d_1, \dots, d_i, \dots, d_n]$  is defined as  $H(D) = \{D\mathbf{a}\}$ , where  $\mathbf{a} = [a_1; \dots; a_i; \dots; a_n]$ . Usually,  $\sum a_i = 1$  is required and  $a_i$  is required to be bounded:

$$H(D) = \{\sum d_i a_i \mid \sum a_i = 1, -\tau_1 \leq a_i \leq \tau_2\} \quad (1)$$

If  $\tau_1 = -inf$  and  $\tau_2 = inf$ ,  $H(D)$  is an affine hull [26]. If  $\tau_1 < 0$  and  $\tau_2 > 0$ ,  $H(D)$  is a reduced affine hull [5]. If

$\tau_1 = 0$  and  $\tau_2 = 1$ ,  $H(\mathbf{D})$  is a convex hull [26]. If  $\tau_1 = 0$  and  $\tau_2 < 1$ ,  $H(\mathbf{D})$  is a reduced convex hull [5].

To rule out the meaningless points which are too far from the sample mean, the regularized affine hull (RAH) [35] is defined as follows to model an image set:

$$H(\mathbf{D}) = \left\{ \sum \mathbf{d}_i a_i \mid \sum a_i = 1, \|\mathbf{a}\|_{l_p} \leq \sigma \right\} \quad (2)$$

## 2.2. Point-to-set distance (PSD)

Given a sample  $\mathbf{x}$  and a set of samples  $\mathbf{D}$ , a point to set distance  $d(\mathbf{x}, \mathbf{D})$  between  $\mathbf{x}$  and  $\mathbf{D}$  can be defined as follows:

$$d(\mathbf{x}, \mathbf{D}) = \|\mathbf{x} - \mathbf{D}\hat{\mathbf{a}}\|_2 \quad (3)$$

where  $\hat{\mathbf{a}} = \arg \min_{\mathbf{a}} \|\mathbf{x} - H(\mathbf{D})\|_2^2$ . When  $H(\mathbf{D})$  is a hull, the solution of  $\min_{\mathbf{a}} \|\mathbf{x} - H(\mathbf{D})\|_2^2$  can be easily obtained by least square regression as  $(\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{x}$  if  $\mathbf{D}^T \mathbf{D}$  is non-singular, or by ridge regression  $(\mathbf{D}^T \mathbf{D} + \lambda \mathbf{I})^{-1} \mathbf{D}^T \mathbf{x}$  if  $\mathbf{D}^T \mathbf{D}$  is (nearly) singular.

To make the PSD more accurate for classification, a projection matrix  $\mathbf{P}$  can be introduced to project the samples into a desired space. The corresponding PSD distance, denoted by  $d_M(\mathbf{x}, \mathbf{D})$ , is then defined as:

$$\begin{aligned} d_M(\mathbf{x}, \mathbf{D}) &= \|\mathbf{P}(\mathbf{x} - \mathbf{D}\hat{\mathbf{a}})\|_2^2 \\ &= (\mathbf{x} - \mathbf{D}\hat{\mathbf{a}})^T \mathbf{P}^T \mathbf{P} (\mathbf{x} - \mathbf{D}\hat{\mathbf{a}}) \\ &= (\mathbf{x} - \mathbf{D}\hat{\mathbf{a}})^T \mathbf{M} (\mathbf{x} - \mathbf{D}\hat{\mathbf{a}}) \end{aligned} \quad (4)$$

where  $\hat{\mathbf{a}} = \arg \min_{\mathbf{a}} \|\mathbf{P}(\mathbf{x} - \mathbf{D}\mathbf{a})\|_2^2$ , and

$$\mathbf{M} = \mathbf{P}^T \mathbf{P}, \quad (5)$$

When  $\hat{\mathbf{a}}$  is obtained, we can form a sample pair  $(\mathbf{x}, \mathbf{D}\hat{\mathbf{a}})$ . Clearly, the PSD  $d_M(\mathbf{x}, \mathbf{D})$  defined in Eq. (4) can be viewed as a Mahalanobis distance [10] between  $\mathbf{x}$  and  $\mathbf{D}\hat{\mathbf{a}}$ , and the matrix  $\mathbf{M}$  is always semi-positive definite.

In PSD based classification, the distance between the query sample  $\mathbf{y}$  and the template set of each class  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_c$  ( $c$  is the number of classes) needs to be computed first. Suppose that the nearest subspace classifier (NSC) is used. Given  $\mathbf{M}$ , for class  $i$ , we have  $\hat{\mathbf{a}}_i = \mathbf{W}_i \mathbf{y}$ , where

$$\mathbf{W}_i = (\mathbf{X}_i^T \mathbf{M} \mathbf{X}_i + \lambda \mathbf{I})^{-1} \mathbf{X}_i^T \mathbf{M}. \quad (6)$$

and then the PSD between  $\mathbf{y}$  and set  $\mathbf{X}_i$  is:

$$d_M(\mathbf{y}, \mathbf{X}_i) = (\mathbf{y} - \mathbf{X}_i \hat{\mathbf{a}}_i)^T \mathbf{M} (\mathbf{y} - \mathbf{X}_i \hat{\mathbf{a}}_i). \quad (7)$$

The class with the minimal PSD is assigned to  $\mathbf{y}$ :  $\text{Label}(\mathbf{y}) = \arg \min_i \{d_M(\mathbf{y}, \mathbf{X}_i)\}$ .

Compared with the nearest convex hull/affine hull classifier (NCH/NAH), which needs to solve  $c$  quadratic programming problems for the query sample  $\mathbf{y}$ , NSC only needs to compute a set of linear projections of  $\mathbf{y}$  with  $\mathbf{W}_i, i = 1, 2, \dots, c$ . Hence, NSC is much more efficient than NCH and NAH.

## 2.3. Set-to-set distance (SSD)

Given two image sets  $\mathbf{D}_1$  and  $\mathbf{D}_2$ , the set-to-set distance (SSD) between them can be defined as follows:

$$d(\mathbf{D}_1, \mathbf{D}_2) = \left\| \mathbf{D}_1 \hat{\mathbf{a}} - \mathbf{D}_2 \hat{\mathbf{b}} \right\|_2^2 \quad (8)$$

where  $\hat{\mathbf{a}}$  and  $\hat{\mathbf{b}}$  can be solved by:

$$(\hat{\mathbf{a}}, \hat{\mathbf{b}}) = \arg \min_{\mathbf{a}, \mathbf{b}} \|H(\mathbf{D}_1) - H(\mathbf{D}_2)\|_2^2 \quad (9)$$

When convex/affine/regularized constraints are imposed on the coefficient vectors  $\mathbf{a}$  and  $\mathbf{b}$ , respectively, the corresponding distances are convex hull based image set distance (CHISD) [5], affine hull based image set distance (AHISD) [5] and regularized nearest points (RNP) [35], respectively. In [35], it has been shown that  $l_2$ -norm regularized affine hull is much faster and can achieve comparable performance to convex/affine/sparse constraints. Given a linear projection matrix  $\mathbf{P}$ , the RNP model is:

$$\begin{aligned} \min_{\mathbf{a}, \mathbf{b}} & \|\mathbf{P}(\mathbf{D}_1 \mathbf{a} - \mathbf{D}_2 \mathbf{b})\|_2^2 + \lambda_1 \|\mathbf{a}\|_2^2 + \lambda_2 \|\mathbf{b}\|_2^2 \\ \text{s.t.} & \sum a_i = 1, \sum b_i = 1 \end{aligned} \quad (10)$$

By solving Eq. (10), the SSD in Eq. (8) becomes:

$$\begin{aligned} d_M(\mathbf{D}_1, \mathbf{D}_2) &= \left\| \mathbf{P}(\mathbf{D}_1 \hat{\mathbf{a}} - \mathbf{D}_2 \hat{\mathbf{b}}) \right\|_2^2 \\ &= (\mathbf{D}_1 \hat{\mathbf{a}} - \mathbf{D}_2 \hat{\mathbf{b}})^T \mathbf{M} (\mathbf{D}_1 \hat{\mathbf{a}} - \mathbf{D}_2 \hat{\mathbf{b}}) \end{aligned} \quad (11)$$

In SSD based classification, given a query image set  $\mathbf{Y}$ , the SSD between it and each template set  $\mathbf{X}_i, i = 1, 2, \dots, c$ , is computed as

$$d_M(\mathbf{Y}, \mathbf{X}_i) = (\mathbf{Y} \hat{\mathbf{a}} - \mathbf{X}_i \hat{\mathbf{b}}_i)^T \mathbf{M} (\mathbf{Y} \hat{\mathbf{a}} - \mathbf{X}_i \hat{\mathbf{b}}_i). \quad (12)$$

$\mathbf{Y}$  can then be classified by  $\text{Label}(\mathbf{Y}) = l(\mathbf{X}_{\hat{i}})$ , where  $\hat{i} = \arg \min_i \{d_M(\mathbf{Y}, \mathbf{X}_i)\}$ .

## 3. Distance metric learning

With the definitions in Section 2, we can then design the metric learning algorithms for PSD and SSD based classification.

### 3.1. Point-to-set distance metric learning (PSDML)

According to Eq. (7), the matrix  $\mathbf{M}$  plays a critical role in the final distance  $d_M(\mathbf{y}, \mathbf{X}_i)$ . It is expected that a good  $\mathbf{M}$  can be learned from the training sample sets  $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_c\}$ , so that the PSD between a query sample  $\mathbf{y}$  and the set  $\mathbf{X}_{l(\mathbf{y})}$  can be reduced, while the PSD between  $\mathbf{y}$  and the other sets  $\mathbf{X}_j, j \neq l(\mathbf{y})$ , can be enlarged, where  $l(\mathbf{y})$  is the label of  $\mathbf{y}$ .

To achieve this goal, with the given training data sets  $\mathbf{X}_i, i = 1, 2, \dots, c$ , we propose the following metric learning

model:

$$\begin{aligned} \min_{\mathbf{M}, \mathbf{a}_{l(\mathbf{x}_i)}, \mathbf{a}_j, \xi_{ij}^N, \xi_i^P, b} & \|\mathbf{M}\|_F^2 + \nu(\sum_{i,j} \xi_{ij}^N + \sum_i \xi_i^P) \\ \text{s.t. } & d_M(\mathbf{x}_i, \mathbf{X}_j) + b \geq 1 - \xi_{ij}^N, j \neq l(\mathbf{x}_i); \\ & d_M(\mathbf{x}_i, \mathbf{X}_{l(\mathbf{x}_i)}) + b \leq -1 + \xi_i^P; \\ & \mathbf{M} \succcurlyeq 0, \forall i, j, \xi_{ij}^N \geq 0, \xi_i^P \geq 0 \end{aligned} \quad (13)$$

where  $\|\cdot\|_F$  denotes the Frobenius norm,  $\mathbf{a}_{l(\mathbf{x}_i)}$  and  $\mathbf{a}_j$  are coefficients vector for  $\mathbf{X}_{l(\mathbf{x}_i)}$  and  $\mathbf{X}_j$ ,  $b$  is the bias and  $\nu$  is a positive constant.  $\xi_i^P$  and  $\xi_{ij}^N$  are slack variables for positive and negative pairs.  $d_M(\mathbf{x}_i, \mathbf{X}_{l(\mathbf{x}_i)})$  is the PSD distance from  $\mathbf{x}_i$  to the set it belongs to (i.e., the PSD of positive pairs), where  $l(\mathbf{x}_i)$  is the class label of  $\mathbf{x}_i$ , and  $d_M(\mathbf{x}_i, \mathbf{X}_j), j \neq l(\mathbf{x}_i)$ , is the PSD from  $\mathbf{x}_i$  to other classes (i.e., the PSD of negative pairs).

Eq. (13) is a joint optimization problem of  $\mathbf{M}$  and  $\{\mathbf{a}_{l(\mathbf{x}_i)}, \mathbf{a}_j\}$ . Like the strategy adopted in many multi-variable optimization problems, we minimize Eq. (13) by optimizing  $\mathbf{M}$  and  $\{\mathbf{a}_{l(\mathbf{x}_i)}, \mathbf{a}_j\}$  alternatively. When  $\mathbf{M}$  is fixed,  $\{\mathbf{a}_{l(\mathbf{x}_i)}, \mathbf{a}_j\}$  are solved for all the training samples. Note that here the “leave-one-out” strategy is used to compute  $\mathbf{a}_{l(\mathbf{x}_i)}$ . That is,  $\bar{\mathbf{X}}_{l(\mathbf{x}_i)}$  is the training sample set of class  $l(\mathbf{x}_i)$  but excluding sample  $\mathbf{x}_i$ . Then the positive pairs are formed as  $(\mathbf{x}_i, \bar{\mathbf{X}}_{l(\mathbf{x}_i)} \hat{\mathbf{a}}_{l(\mathbf{x}_i)})$  and the negative pairs are formed as  $(\mathbf{x}_i, \mathbf{X}_{j, j \neq l(\mathbf{x}_i)} \hat{\mathbf{a}}_{j, j \neq l(\mathbf{x}_i)})$ . We label the negative pair as “+1” and the positive pair is set as “-1”.

Let us denote by  $\mathbf{z}_i = (\mathbf{z}_{i1}, \mathbf{z}_{i2})$  a generated sample pair. The covariance matrix of the two samples in  $\mathbf{z}_i$  is  $\mathbf{C}_i = (\mathbf{z}_{i1} - \mathbf{z}_{i2})(\mathbf{z}_{i1} - \mathbf{z}_{i2})^T$ . Suppose that we generated  $ns$  training sample pairs, and thus we have  $ns$  covariance matrices  $\mathbf{C}_i, i = 1, 2, \dots, ns$ . We label  $\mathbf{C}_i$  as “+1” or “-1” based on the label of  $\mathbf{z}_i$ , and define the following kernel function to measure the similarity between  $\mathbf{C}_i$  and  $\mathbf{C}_j$ :

$$k(\mathbf{C}_i, \mathbf{C}_j) = \text{tr}(\mathbf{C}_i \mathbf{C}_j) = \langle \mathbf{C}_i, \mathbf{C}_j \rangle \quad (14)$$

where  $\text{tr}(\cdot)$  is the trace operator of a matrix and  $\langle \cdot, \cdot \rangle$  means the inner product of matrices.

Suppose that we have a query sample pair, denoted by  $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2)$ . The covariance matrix of  $\mathbf{z}$  is denoted by  $\mathbf{C}$ . We introduce the following discriminative function to judge whether  $\mathbf{z}$  is positive or negative:

$$\begin{aligned} f(\mathbf{C}) &= \sum_i \beta_i l_i k(\mathbf{C}_i, \mathbf{C}) + b \\ &= \sum_i \beta_i l_i \langle \mathbf{C}_i, \mathbf{C} \rangle + b \\ &= \langle \sum_i \beta_i l_i \mathbf{C}_i, \mathbf{C} \rangle + b \end{aligned} \quad (15)$$

where  $l_i$  is the label of pair  $\mathbf{z}_i$ , and  $\beta_i$  is a weight. Let

$$\mathbf{M} = \sum_i \beta_i l_i \mathbf{C}_i. \quad (16)$$

Then we have  $f(\mathbf{C}) = \langle \mathbf{M}, \mathbf{C} \rangle + b$ .

The metric learning problem in Eq. (13) can then be converted into the following problem:

$$\begin{aligned} \min_{\mathbf{M}, b, \xi} & \|\mathbf{M}\|_F^2 + \nu \sum_i \xi_i \\ \text{s.t. } & l_i \langle \mathbf{M}, \mathbf{C}_i \rangle + b \geq 1 - \xi_i, \xi_i \geq 0 \end{aligned} \quad (17)$$

The Lagrange dual problem of the metric learning problem in Eq. (17) is:

$$\begin{aligned} \max_{\beta} & -\frac{1}{2} \sum_{i,j} \beta_i \beta_j l_i l_j k(\mathbf{C}_i, \mathbf{C}_j) + \nu \sum_i \beta_i \\ \text{s.t. } & 0 \leq \beta_i \leq \mu, \sum_i \beta_i l_i = 0 \end{aligned} \quad (18)$$

Obviously, the minimization in Eq. (18) can be easily solved by the support vector machine (SVM) solvers such as LIBSVM [7]. Once  $\beta = [\beta_1, \dots, \beta_i, \dots, \beta_{ns}]$  is obtained by solving Eq. (18),  $\mathbf{M}$  can be obtained by Eq. (16). With  $\mathbf{M}$ , the distance between two samples  $\mathbf{z}_1$  and  $\mathbf{z}_2$  can be computed as:

$$\begin{aligned} d_M(\mathbf{z}_1, \mathbf{z}_2) &= (\mathbf{z}_1 - \mathbf{z}_2)^T \mathbf{M} (\mathbf{z}_1 - \mathbf{z}_2) \\ &= \text{tr}(\mathbf{M} \mathbf{C}) = \langle \mathbf{M}, \mathbf{C} \rangle \end{aligned} \quad (19)$$

If we further require  $d_M(\mathbf{z}_1, \mathbf{z}_2)$  to be a Mahalanobis distance metric,  $\mathbf{M}$  should be semi-positive definite. Similar to Xing et al.’s MMC [33] and Globerson et al.’s MCML [12], we can compute the singular value decomposition (SVD) of  $\mathbf{M} = \mathbf{U} \mathbf{\Lambda} \mathbf{V}$ , where  $\mathbf{\Lambda}$  is the diagonal matrix of eigenvalues, and then set the negative eigenvalues in  $\mathbf{\Lambda}$  to 0, resulting in a new diagonal matrix  $\mathbf{\Lambda}_+$ . Finally, we let  $\mathbf{M}_+ = \mathbf{U} \mathbf{\Lambda}_+ \mathbf{V}$  be the learned matrix.

Once  $\mathbf{M}$  is computed,  $\{\mathbf{a}_{l(\mathbf{x}_i)}, \mathbf{a}_j\}$  are then updated, and the  $\mathbf{M}$  is further updated, and so on. The proposed point-to-set distance metric learning (PSDML) algorithm is summarized in Table 2. The PSDML can be coupled with PSD based classifiers such as NSC [8], NCH [26] and NAH [26] for classification. In this paper, we use NSC since it is much more efficient than NCH and NAH.

Table 2. Algorithm of point to set distance metric learning (PSDML)

Input:  $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_c]$ , label  $l$ ,  $\lambda$  and  $\nu$

Output:  $\mathbf{M}$

- 1 Initialize  $\mathbf{M} = \mathbf{I}$
- 2 While iteration number  $< num$
- 3     Compute  $\mathbf{W}_i, i = 1, \dots, c$  by Eq. (6);
- 4     Construct positive and negative sample pairs;
- 5     Solve Eq. (18) by SVM solver;
- 6     Update  $\mathbf{M}$  by Eq. (16);
- 7 End

### 3.2. Set-to-set distance metric learning (SSDML)

With the SSD defined in Eq. (8), we can also learn a matrix  $\mathbf{M}$  from the training sample sets  $\{\mathbf{X}_1, \dots, \mathbf{X}_i, \dots, \mathbf{X}_n\}$  so that the SSD between sets with the same label can be reduced, while the SSD between sets with different labels can be enlarged. The proposed set-to-set distance metric learning (SSDML) model is formulated as follows:

$$\begin{aligned} \min_{\mathbf{M}, \mathbf{a}_i, \mathbf{a}_j, \mathbf{a}_k, \xi_{ik}^P, \xi_{ik}^N, b} & \|\mathbf{M}\|_F^2 + \nu(\sum_{i,k} \xi_{ik}^P + \sum_{i,j} \xi_{ij}^N) \\ \text{s.t. } & d_M(\mathbf{X}_i, \mathbf{X}_j) + b \geq 1 - \xi_{ij}^N, l(\mathbf{X}_i) \neq l(\mathbf{X}_j); \\ & d_M(\mathbf{X}_i, \mathbf{X}_k) + b \leq -1 + \xi_{ik}^P, l(\mathbf{X}_i) = l(\mathbf{X}_k); \\ & \mathbf{M} \succcurlyeq 0, \forall i, j, k, \xi_{ij}^N \geq 0, \xi_{ik}^P \geq 0 \end{aligned} \quad (20)$$

where  $\mathbf{a}_i, \mathbf{a}_j, \mathbf{a}_k$  are the coefficients vector for image sets  $\mathbf{X}_i, \mathbf{X}_j, \mathbf{X}_k$ ;  $l(\mathbf{X}_i)$  means the label of set  $\mathbf{X}_i$ , and  $\xi_{ik}^P, \xi_{ij}^N$  are the slack variables for positive pairs and negative pairs.

The principles and main procedures of SSDML are similar to the PSDML in Section 3.1. We solve Eq. (20) by optimizing  $M$  and  $\{\mathbf{a}_i, \mathbf{a}_j, \mathbf{a}_k\}$  alternatively. When  $M$  is fixed,  $\{\mathbf{a}_i, \mathbf{a}_j, \mathbf{a}_k\}$  are updated to construct positive and negative sample pairs. When the sample pairs are given, the updating of matrix  $M$  can also be converted into the problem in Eq. (17). The algorithm of SSDML is summarized in Table 3. Note that the work in [32] relies on CHISD [5] and SANP [16]. As RNP [35] is much faster than convex/sparse hull based SSD computation, we choose it to learn the Mahalanobis distance metric based on  $l_2$ -norm regularized affine hull.

Table 3. Algorithm of set to set distance metric learning (SSDML)

Input: Training image sets $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n]$ , label $l, \lambda_1, \lambda_2$ and $\nu$	
Output: $M$	
1	Initialize $M = I$
2	While iteration number $< num$
3	Compute SSD for each image set $\mathbf{X}_i$ by Eq. (10);
4	Construct positive and negative sample pairs;
5	Solve Eq. (18) by SVM solver;
6	Update $M$ by Eq. (16);
7	End

### 3.3. Discussions

There are close relationships between the proposed PSDML/SSDML and SVM. The geometric interpretation of  $\nu$ -SVM is to find the closest points in two (reduced) convex hulls [4]. Given two classes  $\mathbf{X}_1$  and  $\mathbf{X}_2$ , the SVM is to solve the following problem [4]:

$$\begin{aligned} \min & \|\mathbf{X}_1 \mathbf{a}_1 - \mathbf{X}_2 \mathbf{a}_2\|_2^2 \\ \text{s.t.} & \sum a_{1i} = 1, \sum a_{2j} = 1, 0 \leq a_{1i}, a_{2j} \leq \mu \end{aligned} \quad (21)$$

It can be easily found that the associated discrimination function of SVM is  $f(\mathbf{y}) = \mathbf{w}^T \mathbf{y} + \mathbf{b}$ , where  $\mathbf{w} = (\mathbf{X}_1 \mathbf{a}_1 - \mathbf{X}_2 \mathbf{a}_2)/2$ ,  $p = (\mathbf{X}_1 \mathbf{a}_1 + \mathbf{X}_2 \mathbf{a}_2)/2$ ,  $\mathbf{b} = -\mathbf{w}^T \mathbf{p} = (\mathbf{a}_2^T \mathbf{X}_2^T \mathbf{X}_2 \mathbf{a}_2 - \mathbf{a}_1^T \mathbf{X}_1^T \mathbf{X}_1 \mathbf{a}_1)/4$ .

Then we have the following observation:

$$\begin{aligned} f(\mathbf{y}) &= \mathbf{w}^T \mathbf{y} + \mathbf{b} \\ &= \frac{(\mathbf{X}_1 \mathbf{a}_1 - \mathbf{X}_2 \mathbf{a}_2)^T}{2} \mathbf{y} + \frac{\mathbf{a}_2^T \mathbf{X}_2^T \mathbf{X}_2 \mathbf{a}_2 - \mathbf{a}_1^T \mathbf{X}_1^T \mathbf{X}_1 \mathbf{a}_1}{4} \\ &= \frac{\|\mathbf{y} - \mathbf{X}_2 \mathbf{a}_2\|_2^2 - \|\mathbf{y} - \mathbf{X}_1 \mathbf{a}_1\|_2^2}{4} \\ &= \frac{d(\mathbf{y}, \mathbf{X}_2) - d(\mathbf{y}, \mathbf{X}_1)}{4} \end{aligned} \quad (22)$$

Hence, similar to PSD based classification, the discriminative function of SVM actually uses the distance between the test sample  $\mathbf{y}$  and each class. If  $f(\mathbf{y}) \geq 0$ , then  $\mathbf{y}$  belongs to the first class. If  $f(\mathbf{y}) < 0$ , then  $\mathbf{y}$  belongs to

the second class. The difference, however, lies in that PSD based classifiers (e.g., NSC, NCH and NAH) solve  $\mathbf{a}_1$  and  $\mathbf{a}_2$  for each test sample while SVM learns  $\mathbf{a}_1$  and  $\mathbf{a}_2$  from the training set by classification loss minimization and margin maximization. The conventional PSD based classifiers ignore the training label information in computing  $\mathbf{a}_1$  and  $\mathbf{a}_2$ . With metric learning, PSDML can further utilize the class label to learn a discriminative metric for the point-to-set distance, and thus may result in better classification performance.

For set based classification, SVM can not be directly used. Actually, given two sets, SVM considers each set as one class and the distance between two classes is used as the SSD, which corresponds to CHISD [5]. Hence, it still ignores the discriminative information in calculating SSD, and is essentially different from the proposed SSD metric learning method.

Additionally, we formulate both PSDML and SSDML as a sample pair classification problem, which can be solved by standard SVM solvers. This makes metric learning very efficient.

## 4. Experimental result and analysis

We verify the performance of PSDML and SSDML on various visual classification tasks. In Section 4.1, we test PSDML on gender classification, digit recognition, object categorization and face recognition, while in Section 4.2, we test SSDML on video-to-video based face recognition.

### 4.1. PSDML experiments

#### 4.1.1 Parameter setting and competing methods

There are two parameters in PSDML, i.e.,  $\lambda$  in Eq. (6) and  $\nu$  in Eq. (17). For SSDML, there are three parameters, i.e.,  $\lambda_1$  and  $\lambda_2$  in Eq. (10) and  $\nu$  in Eq. (17). For both PSDML and SSDML,  $\nu$  in Eq. (17) is set to the default value 1 in LIBSVM. For PSDML,  $\lambda$  is chosen by cross-validation on the training set. For SSDML,  $\lambda_1$  and  $\lambda_2$  are fixed as 0.001 and 0.1, respectively.

We compare PSDML with four state-of-the-art metric learning methods (LMNN [31], ITML [10], NCA [13] and MCML [12]), three PSD based classifiers (NSC [8], NCH [26] and NAH [26]), the classical nearest neighbor classifier (NNC) and SVM. The Matlab source codes of LMNN, ITML, NCA, and MCML are obtained from the original authors, and we used the SVM toolbox from [7]. We implemented NNC, NCH, NAH and NSC. The parameters of the competing methods are tuned for their best results.

#### 4.1.2 Gender classification

A non-occluded subset (14 images per subject) of the AR dataset [22] is used, which consists of 50 male and 50 female subjects. We use the images from the first 25 males

and 25 females for training, and the remaining images for testing. The images were cropped to  $60 \times 43$ . PCA was used to reduce the dimension of each image to 30 and 50, respectively. The experimental results listed in Table 4 show that PSDML gets the highest accuracy and improves the performance of PSD based classifiers (NSC, NCH and NAH).

dim.	NN	NSC	NCH	NAH	SVM
30	90.6	92.1	91.1	91.7	92.1
50	90.3	93.3	91.4	84.3	91.0
dim.	LMNN	ITML	NCA	MCML	PSDML
30	91.3	90.8	91.4	90.7	<b>93.7</b>
50	91.0	90.7	91.4	92.1	<b>95.4</b>

### 4.1.3 Digit recognition

Three handwritten digit datasets, Semeion [2], USPS [17] and MNIST [21], are used here.

**Semeion:** The Semeion dataset [2] has 1,593 handwritten digits from around 80 persons. Each sample is a  $16 \times 16$  binarized image. The recognition rate on the raw features is shown in Table 5. On this dataset, the performance of NSC is much better than NNC. PSDML gets a recognition accuracy of 95.9%, which is the highest among all the methods used in comparison.

dim.	NN	NSC	NCH	NAH	SVM
256	91.4	94.2	94.1	92.5	93.4
dim.	LMNN	ITML	NCA	MCML	PSDML
256	93.9	93.5	93.9	90.0	<b>95.9</b>

**USPS:** The USPS dataset includes 7,291 training and 2,007 testing images [17]. Each sample is a  $16 \times 16$  image. The experimental results on three dimensions (100, 150, 256) are shown in Table 6. We see that the results of NNC and NSC are similar. PSDML achieves the highest accuracy on different dimensions and its performance is comparable to other state-of-the-art metric learning methods.

dim.	NN	NSC	NCH	NAH	SVM
100	94.9	94.3	88.2	91.8	92.3
150	94.8	94.5	89.3	91.9	92.7
256	94.6	94.3	89.7	91.8	92.7
dim.	LMNN	ITML	NCA	MCML	PSDML
100	95.2	95.0	95.1	95.2	<b>95.4</b>
150	95.2	95.1	95.0	95.1	<b>95.3</b>
256	95.0	94.9	94.8	94.9	<b>95.2</b>

**MNIST:** The MNIST [21] dataset contains a training set of 60,000 samples and a test set of 10,000 samples. There

are 10 classes of images, and the size of each image is  $28 \times 28$ . We randomly select 200 samples per class for training and the image dimension is reduced to 100 by PCA. Ten random experiments are conducted and the average recognition rate is shown in Table 7. Again, PSDML performs the best among all methods.

dim.	NN	NSC	NCH	NAH	SVM
100	93.3	95.2	96.0	94.0	95.7
dim.	LMNN	ITML	NCA	MCML	PSDML
100	95.0	93.4	93.5	90.1	<b>96.3</b>

### 4.1.4 Object categorization

The 17 category OXFORD flower dataset [23] is used. It contains 17 species of flowers with 80 images for each class. The  $\chi^2$  distance matrices of seven features (i.e., HSV, HOG, SIFTint, SIFTbdy, color, shape and texture vocabularies) are directly used as the input and the experiments are conducted based on the three predefined training, validation, and test splits. We test the performance of PSDML on each feature and the results are shown in Table 8. From the results we see that PSDML achieves the highest accuracy on all the seven features.

Table 8. Accuracy (%) on the 17 category OXFORD flowerers

Features	NN	NSC	NAH	NAH	SVM
Color	52.3±2.2	55.4±2.7	55.2±2.8	56.3±2.8	56.9±2.6
Shape	53.7±3.5	66.5±2.1	66.7±2.0	63.4±1.3	60.0±2.9
Texture	31.9±3.6	52.4±2.1	52.4±1.5	45.5±1.8	47.8±3.4
HSV	52.0±2.6	59.2±2.3	59.4±2.3	57.2±3.5	57.0±2.9
HOG	36.9±1.7	51.6±2.5	51.8±2.9	47.6±2.6	47.3±1.9
SIFTint	58.7±2.1	66.5±1.3	66.5±1.4	64.5±1.0	59.7±1.0
SIFTbdy	51.7±0.9	57.6±2.3	57.7±2.2	57.6±2.8	47.5±2.8
Features	LMNN	ITML	NCA	MCML	ISDML
Color	53.1±2.5	53.5±2.6	52.8±2.8	54.1±2.7	<b>58.8±4.0</b>
Shape	50.1±1.0	55.0±1.4	54.5±2.0	55.5±1.5	<b>67.8±2.0</b>
Texture	35.5±3.0	36.2±2.5	33.8±2.6	34.5±2.0	<b>55.0±1.3</b>
HSV	54.8±2.7	53.5±3.0	54.0±2.9	52.9±3.1	<b>61.6±3.2</b>
HOG	38.3±1.1	37.5±2.5	38.2±2.5	38.7±2.8	<b>55.0±5.9</b>
SIFTint	60.0±3.4	61.2±1.9	59.8±1.5	60.4±1.3	<b>69.1±1.8</b>
SIFTbdy	53.3±4.1	54.2±2.5	53.3±2.9	53.3±2.1	<b>60.6±4.0</b>

### 4.1.5 Face recognition

We then test the performance of PSDML on face recognition. As in [31], the Extended Yale B database [11] is used here. In addition, the FERET database [25] is also used since the images have huge pose variations, making it a good test-bed for metric learning methods.

**Extended YaleB:** The Extended YaleB database contains 2,414 frontal face images of 38 persons [11]. There are about 64 images for each subject. The original images were

cropped to  $192 \times 168$  pixels. This database has varying illuminations and expressions. A randomly generated matrix from a zero-mean normal distribution is used to project the face image onto a 504-dimensional vector. We randomly choose 15 samples per subject for training and the rest images are used for test. PCA is used to reduce the dimension to 50, 100 and 150, respectively. On this database, the performance of NSC is much better than NNC. Compared with NSC, PSDML improves the recognition rate by about 4% and it works much better than other competing methods.

Table 9. Accuracy (%) on the Extended YaleB database

dim.	NN	NSC	NCH	NAH	SVM
50	76.3	86.1	70.9	86.1	78.1
100	80.2	88.2	75.5	87.6	82.4
150	78.3	88.9	77.1	88.9	82.3
dim.	LMNN	ITML	NCA	MCML	ISDML
50	77.4	78.3	78.9	79.0	<b>90.0</b>
100	81.1	81.0	82.4	82.9	<b>92.2</b>
150	81.8	83.1	83.5	82.1	<b>93.0</b>

**FERET:** The FERET face database is a large and popular database for evaluating state-of-the-art face recognition algorithms [25]. We use a subset of the database that includes 1,400 images from 200 individuals (each has 7 images). It consists of the images whose names are marked with two character strings: “ba”, “bj”, “bk”, “bd”, “be”, “bf”, “bg”. This subset involves variations in facial expression, illumination, and pose. The facial portion of each image was automatically cropped based on the location of eyes and mouth, and the cropped image was resized to  $60 \times 50$  pixels and further pre-processed by histogram equalization.

We randomly select four images per subject as the training set and the remaining images are used as the test set. The recognition rates are shown in Table 10. In this dataset, the performance of NSC is worse than NNC. This is because there are great pose variations in this subset, and thus using hull to model the image set is not suitable. By metric learning, however, the classification rate can be improved greatly. The result of PSDML is much better than LMNN, ITML, NCA and MCML, which validates the effectiveness of our algorithm.

#### 4.1.6 Time comparison

To show the efficiency of PSDML, we compare the training time of different metric learning methods. All algorithms are run in an Intel(R) Core(TM) i7- 2600K (3.4GHz) PC. The average training time on the MNIST dataset is listed in Table 11. We see that PSDML is much faster than other metric learning methods. In particular, it is nearly 500 times faster than MCML.

Table 10. Accuracy (%) on the FERET

dim.	NN	NSC	NCH	NAH	SVM
50	40.5	38.9	37.6	38.9	45.8
100	48.0	42.4	41.5	42.4	59.5
150	48.8	43.7	42.6	43.7	64.6
dim.	LMNN	ITML	NCA	MCML	PSDML
50	60.0	61.5	59.5	60.5	<b>64.0</b>
100	62.7	63.8	61.6	63.3	<b>67.8</b>
150	63.5	64.8	62.0	64.5	<b>67.8</b>

Table 11. Training time (s) on the MNIST

Methods	LMNN	ITML	NCA	MCML	PSDML
run_time	75.9	141.0	3885.1	11825.1	<b>24.7</b>

## 4.2. SSDML experiments

We then test SSDML for set-to-set based classification tasks. The benchmark YouTube Celebrities dataset is used. In this experiment, we compare SSDML with those SSD based classification methods (CHISD [5], AHISD [5], SANP [16], RNP [35], MMD [29] and MDA [28]) and set-to-set similarity based methods (MSM [34] and DCC [20]). The source codes of these methods are from the original authors and we tune the parameters for their best results.

The Youtube Celebrities [20] is a large scale video dataset for face tracking and recognition, consisting of 1,910 video sequences of 47 celebrities collected from YouTube. As the videos were captured in unconstrained environments, the recognition task becomes much more challenging due to large variations in pose, illumination and expressions. The face in each frame is detected by the Viola-Jones face detector and resized to a  $30 \times 30$  grayscale image.

The intensity value is used as feature. Three video sequences per subject are selected for training and six for testing. Five-fold cross validation is used. The experiments for 50, 100, 200 frames per set are conducted. The result is shown in Table 12. We can see that SSDML outperforms all the other methods on different frames per set.

Table 12. Recognition rates on YouTube (%)

Methods	50	100	200
MSM [34]	54.8±8.7	57.4±7.7	56.7±6.9
DCC [20]	57.6±8.0	62.7±6.8	65.7±7.0
MMD [29]	57.8±6.6	62.8±6.2	64.7±6.3
MDA [28]	58.5±6.2	63.3±6.1	65.4±6.6
AHISD [5]	57.5±7.9	59.7±7.2	57.0±5.5
CHISD [5]	58.0±8.2	62.8±8.1	64.8±7.1
SANP [16]	57.8±7.2	63.1±8.0	65.6±7.9
RNP [35]	59.9±7.3	63.3±8.1	64.4±7.8
SSDML	<b>61.9±7.3</b>	<b>65.0±8.1</b>	<b>67.0±7.1</b>

## 5. Conclusion

We extended the point-to-point distance metric learning to point-to-set distance metric learning (PSDML) and set-to-set distance metric learning (SSDML). Positive and negative sample pairs were generated from training sample sets by computing point-to-set distance (PSD) and set-to-set distance (SSD). Each sample pair was represented by its covariance matrix and a covariance kernel based discrimination function was proposed for sample pair classification. Finally, we showed that the proposed metric learning problem can be efficiently solved by SVM solvers. Experiments on various visual classification problems demonstrated that the proposed PSDML and SSDML methods can effectively improve the performance of PSD and SSD based classification. Compared with the state-of-the-art metric learning methods such as LMNN, ITML and MCML, the proposed method can achieve better classification accuracy and is significantly faster in training.

## Acknowledgments

This work is partly supported by the HK RGC GRF grant (PolyU 5313/12E) and by NSFC under Grant 61271093.

## References

- [1] V. Ablavsky and S. Sclaroff. Learning parameterized histogram kernels on the simplex manifold for image and action classification. In *ICCV 2011*.
- [2] A. Asuncion and D. J. Newman. Uci machine learning repository, 2007.
- [3] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *ICML 2004*.
- [4] D. Burges. A geometric interpretation of  $\nu$ -svm classifiers. *NIPS 2000*.
- [5] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *CVPR 2010*.
- [6] H. Cevikalp, B. Triggs, and R. Polikar. Nearest hyperdisk methods for high-dimensional classification. In *ICML 2008*.
- [7] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.
- [8] J.-T. Chien and C.-C. Wu. Discriminant waveletfaces and nearest feature classifiers for face recognition. *TPAMI*, 24(12):1644–1649, 2002.
- [9] R. G. Cinbis, J. Verbeek, and C. Schmid. Unsupervised metric learning for face identification in tv video. In *ICCV 2011*.
- [10] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML 2007*.
- [11] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *TPAMI*, 23(6):643–660, 2001.
- [12] A. Globerson and S. Roweis. Metric learning by collapsing classes. In *NIPS 2006*.
- [13] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *NIPS 2004*.
- [14] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *ICCV 2009*.
- [15] M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *ECCV 2010*.
- [16] Y. Hu, A. S. Mian, and R. Owens. Sparse approximated nearest points for image set classification. In *CVPR 2011*.
- [17] J. J. Hull. A database for handwritten text recognition research. *TPAMI*, 16(5):550–554, 1994.
- [18] N. Jiang, W. Liu, and Y. Wu. Order determination and sparsity-regularized metric learning adaptive visual tracking. In *CVPR 2012*.
- [19] D. Kedem, S. Tyree, K. Weinberger, F. Sha, and G. Lanckriet. Non-linear metric learning. In *NIPS 2012*.
- [20] T. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *TPAMI*, 29(6):1005–1018, 2007.
- [21] Y. LeCun and C. Cortes. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>.
- [22] A. Martinez. The ar face database. *CVC Technical Report*, 24, 1998.
- [23] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *CVPR 2006*.
- [24] S. Parameswaran and K. Weinberger. Large margin multi-task metric learning. In *NIPS 2010*.
- [25] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *TPAMI*, 22(10):1090–1104, 2000.
- [26] P. Vincent and Y. Bengio. K-local hyperplane and convex distance nearest neighbor algorithms. *NIPS 2002*.
- [27] F. Wang, W. Zuo, L. Zhang, D. Meng, and D. Zhang. A kernel classification framework for metric learning. *arXiv:1309.5823*, 2013.
- [28] R. Wang and X. Chen. Manifold discriminant analysis. In *CVPR 2009*.
- [29] R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image set. In *CVPR 2008*.
- [30] Z. Wang, Y. Hu, and L.-T. Chia. Image-to-class distance metric learning for image classification. In *ECCV 2010*.
- [31] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS 2006*.
- [32] Y. Wu, M. Minoh, M. Mukunoki, and S. Lao. Set based discriminative ranking for recognition. In *ECCV 2012*.
- [33] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS 2002*.
- [34] O. Yamaguchi, K. Fukui, and K. Maeda. Face recognition using temporal image sequence. In *FG 1998*.
- [35] M. Yang, P. Zhu, L. Van Gool, and L. Zhang. Face recognition based on regularized nearest points between image sets. In *FG 2013*.
- [36] G.-T. Zhou, T. Lan, W. Yang, and G. Mori. Learning class-to-image distance with object matchings.